Replace the paragraph on page 4, lines 1-12 with the following:

In one embodiment of the invention, as illustrated by Equation (6), a hashing of a message "m" is performed by summing the message string with a key string "a" and then forming the square of that summation. A modular "p" operation performed on the result of the squaring operation and a modular 2^l operation is performed on the result of the modular "p" operation. In this case, both "m" and "a" are of the same length, that is, "n" bits or "w" words long. It should be noted that "a" may be longer than "n" bits, but "n" bits is preferable. The value "l" refers to the length in bits of the shortened string that results from the hashing. The value "p" is selected as the first prime number greater than 2^n where "n" is the number of bits in the message string "m". It should be noted that Equation (6) provides a hashing method that satisfies Equations (1) and (2), that is, the hashing method of Equation (6) is Δ universal.

Replace the paragraph beginning at page 6, line 17 and ending at page 7, line 10 with the following:

FIG. 3 illustrates a method for performing the ε Δ universal hashing method described by Equation (8). In step 170 index "i" is set equal to 1 and the variable SUM is set equal to 0. In step 172 the value of "k" is inputted. "k" is equal to the number of strings or messages that will be inputted to produce a single shortened message. In step 174 message or string m_i is separated, and in step 176 input key a_i is inputted. It should be noted that message or string m_i and input key a_i are of equal length and have "n" bits composing "w" words. Key "a_i" is a random or pseudo-random number and may be longer than "n" bits, but "n" bits is preferable. Preferably, a_i is a random number. Random numbers can be generated from many sources such as pseudo-random generators. In step 178 sum s_i is formed by forming the sum of message m_i and key a_i. In step 180 the square of s_i is set equal to variable SQ_i. In step 182 the variable SUM is set equal to the variable SUM plus SQ_i. In step 184 the value of "i" is checked to determine if it is equal to the value "k". If it is not equal to the value "k", step 186 is executed where the value of index "i" is incremented by "l" and then step 174 is executed. If in step 184 the value of "i" is determined to be equal to "k", step 188 is executed where a modular "p" operation is performed on the current value of the variable SUM. As discussed previously, the value "p" is the next prime number greater than the value 2^n; however, "p" may be a larger prime which may degrade performance. In step 190 a modular 2^l operation is